Measuring Performance

- The only important question: “how fast will my program run?”

- CPU execution time for a program
  - = CPU clock cycles * cycle time

- In computer design, trade-off between:
  - clock cycle time, and
  - number of cycles required for a program

Cycles Per Instruction

- The execution time of a program clearly must depend on the number of instructions
  - but different instructions take different times

- An expression that includes this is:-
  - CPU clock cycles = I_c * CPI
    - I_c = instructions count
    - CPI = average clock cycles per instruction

Performance and speed

- Performance for a program on a particular machine

$$\text{Performance}(x) = \frac{1}{\text{Execution }(x)}$$

- X is n time faster than y

$$\frac{\text{Performance}(x)}{\text{Performance }(y)} = \frac{\text{Execution }(y)}{\text{Execution }(x)} = n$$

Purchasing Decision

- Computer A has a 100MHz processor
- Computer B has a 200MHz processor
- So, B is faster ? No

Computer A

- clock cycle time = 1/f =10 ns / cycle
- CPI = 0.5 for program X

- CPU clock cycles (A) = Ic * 0.5

- CPU time(A) = CPU clock cycles *clock cycle time
  = Ic * 0.5 * 10 n= 5 Ic ns

Computer B

- clock cycle time = 1/f
  =5 ns / cycle
- CPI = 2 for program X

•CPU clock cycles (B) = Ic * 2

•CPU time (B) = CPU clock cycles * clock cycle time
= Ic * 2 * 5 n = 10 Ic ns

Performance (A) / Performance (B)= 2

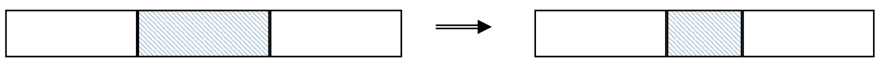
So, A is better than B

Amdahl's Law

Speedup due to enhancement E:

$$SPeedup(E) = \frac{\text{Performance with enhancment}}{\text{Performance without enhancment}}$$

$$SPeedup(E) = \frac{\text{Execution without enhancment}}{\text{Execution with enhancment}}$$



Suppose that enhancement E accelerates a fraction F of the task by a factor S and the remainder of the task is unaffected then,

$$\text{ExTime with } E = \left[(1-F) + \frac{F}{S}\right] * \text{ ExTime without } E$$

$$SPeedup(E) = \frac{1}{\left[(1-F) + \frac{F}{S}\right]}$$

Examples

- Example 1: If 90% of a program can be parallelized (i.e., 10% must be executed sequentially), then the maximum speed-up which can be achieved on 5 processors is

$$\text{Speedup}(E) = \frac{1}{(1-0.9)+\frac{0.9}{5}} = \frac{1}{0.1+0.18} = \frac{1}{0.28} \approx 3.6$$

(the program can theoretically run 3.6 times faster on five processors than on one)

Examples

- Example 2: if the part that can be improved is 30% of the overall system and its performance can be doubled for a system.

$$\text{Speedup}(E) = \frac{1}{(1-0.3)+\frac{0.3}{2}} = \frac{1}{0.7+0.15} = \frac{1}{0.85} \approx 1.18$$

Explanation:

- $F = 0.3$: fraction of the system that is improved
- $S = 2$: speed-up factor for that portion
- $1 - F = 0.7$: fraction unaffected by the improvement

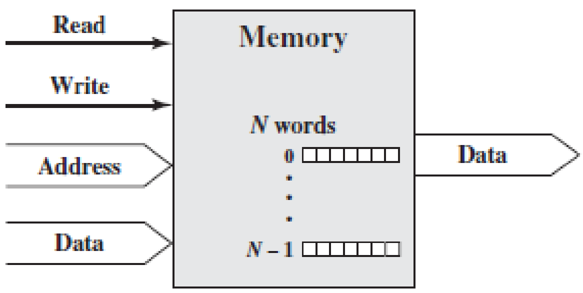
Interconnection Structures

- A computer has **three main modules:**

  Processor, Memory, and I/O.
- These modules **communicate** through connection paths.
- The set of these paths is the **Interconnection Structure.**
- Its **design** depends on the type of data exchange among modules.

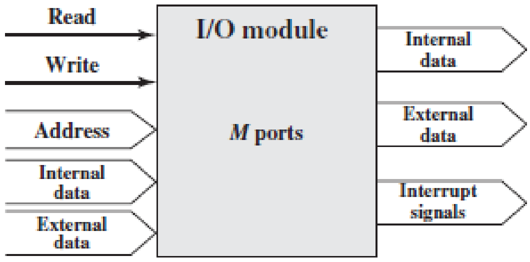
Memory Connection

- Receives and sends data
- Receives addresses (of locations)
- Receives control signals
  - Read
  - Write
  - Timing



Input/Output Connection

- Output
  - Receive data from computer
  - Send data to peripheral
- Input
  - Receive data from peripheral
  - Send data to computer
- Receive control signals from computer
- Send control signals to peripherals
  - e.g. spin disk
- Receive addresses from computer
  - e.g. port number to identify peripheral
- Send interrupt signals (control)



CPU Connection

- Reads instruction and data
- Writes out data (after processing)
- Sends control signals to other units
- Receives (& acts on) interrupts
- Sends address